Velib-2014

Matias BOTTARINI

2025-06-05

Vélib data

```
data <- get(load('velib.Rdata'))
img_height = 600
img_width = 800
```

EDA

For each of the 1189 stations we have the GPS position, bonus (integer) and 181 variables on each station. We think that the 181 variables are the hourly states on each station. The state goes from 0 to 1, so we think that maybe it is the availability of bikes in the station.

Here the summary of the first 10 stations.

```
Xsum = t(data[["data"]])
summary(Xsum[,1:10])
```

```
##      19117             17111              6103             15042
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.05455   Min.   :0.00000
##  1st Qu.:0.07143   1st Qu.:0.08696   1st Qu.:0.32727   1st Qu.:0.09524
##  Median :0.15385   Median :0.43478   Median :0.42593   Median :0.38095
##  Mean   :0.27025   Mean   :0.48308   Mean   :0.44742   Mean   :0.46416
##  3rd Qu.:0.46429   3rd Qu.:0.86957   3rd Qu.:0.55556   3rd Qu.:0.88889
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##      12003             13038            17041             41203
##  Min.   :0.05882   Min.   :0.0000   Min.   :0.05128   Min.   :0.0000
##  1st Qu.:0.32075   1st Qu.:0.1250   1st Qu.:0.23077   1st Qu.:0.1373
##  Median :0.60000   Median :0.2500   Median :0.35897   Median :0.3725
##  Mean   :0.56110   Mean   :0.3347   Mean   :0.39776   Mean   :0.3808
##  3rd Qu.:0.79412   3rd Qu.:0.5833   3rd Qu.:0.53846   3rd Qu.:0.6471
##  Max.   :1.00000   Max.   :0.9583   Max.   :0.94872   Max.   :0.8235
##      43401             5015
##  Min.   :0.3556   Min.   :0.0000
##  1st Qu.:0.4889   1st Qu.:0.1905
##  Median :0.6000   Median :0.3387
##  Mean   :0.6422   Mean   :0.4727
##  3rd Qu.:0.7500   3rd Qu.:0.9048
##  Max.   :1.0000   Max.   :1.0000
```

Let's plot the hourly data for station 1. We can see that during the week, the availability of bikes are 0 when we are close to the weekends (first 24h) and last 48 h, and also at the end of the days (evenings) during week days.

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
png(width = img_width, height = img_height,'station-19117.png')
h = seq(ymd_h("2014-08-31-11"), ymd_h("2014-09-07-23"), by = "hours")

df = as.data.frame(Xsum)
ggplot(data=df, aes(x=h, y=`19117`)) +
  labs (y = "19117 capacity") +
  geom_line() +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%a")
dev.off()
```

```
## pdf
##   2
```

Let's have a look at the GPS location of each Velib station. We can see that all are instlaled in Paris.

```
#install.packages("leaflet")
library(leaflet)
palette = colorFactor("RdYlBu", domain = NULL)
leaflet(data[["position"]]) %>% addTiles() %>%
addCircleMarkers(radius = 3,
stroke = FALSE, fillOpacity = 0.9)
```

Data visualization (PCA)

Let's use PCA on the data. We observe that after component 19 we have more than 90% of the information.

```
X =data[["data"]]
pca = princomp(X)
summary(pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     2.94595 2.2256993 1.02580832 0.92939356 0.87593862
## Proportion of Variance 0.40575 0.2316012 0.04919718 0.04038379 0.03587196
## Cumulative Proportion  0.40575 0.6373512 0.68654837 0.72693216 0.76280412
##                           Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     0.6317361 0.60541555 0.56374901 0.51267364 0.47822454
## Proportion of Variance 0.0186586 0.01713621 0.01485865 0.01228824 0.01069231
## Cumulative Proportion  0.7814627 0.79859893 0.81345758 0.82574582 0.83643813
##                           Comp.11     Comp.12     Comp.13     Comp.14
## Standard deviation     0.46299693 0.443055795 0.406931634 0.386061822
## Proportion of Variance 0.01002222 0.009177504 0.007741955 0.006968214
## Cumulative Proportion  0.84646035 0.855637855 0.863379810 0.870348024
##                           Comp.15     Comp.16     Comp.17     Comp.18
## Standard deviation     0.377796432 0.376243427 0.354901211 0.349330399
## Proportion of Variance 0.006673036 0.006618287 0.005888744 0.005705327
## Cumulative Proportion  0.877021059 0.883639346 0.889528091 0.895233417
```

```
##                            Comp.19     Comp.20     Comp.21     Comp.22
## Standard deviation     0.330714512 0.312363821 0.290256192 0.279110479
## Proportion of Variance 0.005113453 0.004561726 0.003938862 0.003642168
## Cumulative Proportion  0.900346870 0.904908596 0.908847458 0.912489626
##                            Comp.23     Comp.24     Comp.25     Comp.26
## Standard deviation     0.268512977 0.261121155 0.248672422 0.243156650
## Proportion of Variance 0.003370841 0.003187806 0.002891099 0.002764267
## Cumulative Proportion  0.915860467 0.919048273 0.921939372 0.924703640
##                            Comp.27     Comp.28     Comp.29     Comp.30
## Standard deviation     0.237264997 0.228302278 0.223770452 0.217401416
## Proportion of Variance 0.002631935 0.002436847 0.002341064 0.002209696
## Cumulative Proportion  0.927335574 0.929772421 0.932113485 0.934323181
##                            Comp.31     Comp.32     Comp.33     Comp.34
## Standard deviation     0.208576385 0.204578833 0.199507628 0.196939638
## Proportion of Variance 0.002033939 0.001956722 0.001860916 0.001813318
## Cumulative Proportion  0.936357120 0.938313842 0.940174758 0.941988076
##                            Comp.35     Comp.36     Comp.37     Comp.38
## Standard deviation     0.190072024 0.186770482 0.184017194 0.177732700
## Proportion of Variance 0.001689056 0.001630888 0.001583159 0.001476871
## Cumulative Proportion  0.943677133 0.945308021 0.946891180 0.948368050
##                            Comp.39     Comp.40     Comp.41     Comp.42
## Standard deviation     0.174447155 0.166038880 0.165230754 0.161548669
## Proportion of Variance 0.001422773 0.001288924 0.001276408 0.001220154
## Cumulative Proportion  0.949790823 0.951079747 0.952356155 0.953576309
##                            Comp.43    Comp.44     Comp.45     Comp.46
## Standard deviation     0.157359122 0.15542404 0.153792366 0.152869407
## Proportion of Variance 0.001157688 0.00112939 0.001105802 0.001092569
## Cumulative Proportion  0.954733997 0.95586339 0.956969189 0.958061758
##                            Comp.47     Comp.48      Comp.49      Comp.50
## Standard deviation     0.148939695 0.148570875 0.1460866605 0.1439313468
## Proportion of Variance 0.001037119 0.001031989 0.0009977664 0.0009685421
## Cumulative Proportion  0.959098877 0.960130867 0.9611286330 0.9620971751
##                             Comp.51      Comp.52      Comp.53      Comp.54
## Standard deviation     0.1390159053 0.1379226793 0.1359322259 0.1339224099
## Proportion of Variance 0.0009035178 0.0008893631 0.0008638784 0.0008385216
## Cumulative Proportion  0.9630006929 0.9638900561 0.9647539344 0.9655924560
##                             Comp.55      Comp.56      Comp.57      Comp.58
## Standard deviation     0.1313632637 0.1297716417 0.1270937156 0.1259762804
## Proportion of Variance 0.0008067809 0.0007873491 0.0007551894 0.0007419682
## Cumulative Proportion  0.9663992369 0.9671865860 0.9679417755 0.9686837437
##                             Comp.59      Comp.60      Comp.61      Comp.62
## Standard deviation     0.1238803368 0.1217550325 0.1205850195 0.1183895238
## Proportion of Variance 0.0007174845 0.0006930772 0.0006798208 0.0006552911
## Cumulative Proportion  0.9694012282 0.9700943053 0.9707741261 0.9714294173
##                            Comp.63      Comp.64      Comp.65      Comp.66
## Standard deviation     0.117380887 0.1160706994 0.1151703530 0.1127350405
## Proportion of Variance 0.000644173 0.0006298729 0.0006201391 0.0005941904
## Cumulative Proportion  0.972073590 0.9727034632 0.9733236023 0.9739177927
##                             Comp.67      Comp.68      Comp.69      Comp.70
## Standard deviation     0.1113493010 0.1103172769 0.1098202067 0.1081682408
## Proportion of Variance 0.0005796725 0.0005689771 0.0005638613 0.0005470251
## Cumulative Proportion  0.9744974652 0.9750664424 0.9756303036 0.9761773288
##                             Comp.71      Comp.72      Comp.73      Comp.74
## Standard deviation     0.1064078420 0.1055582784 0.1041190511 0.1037252808
```

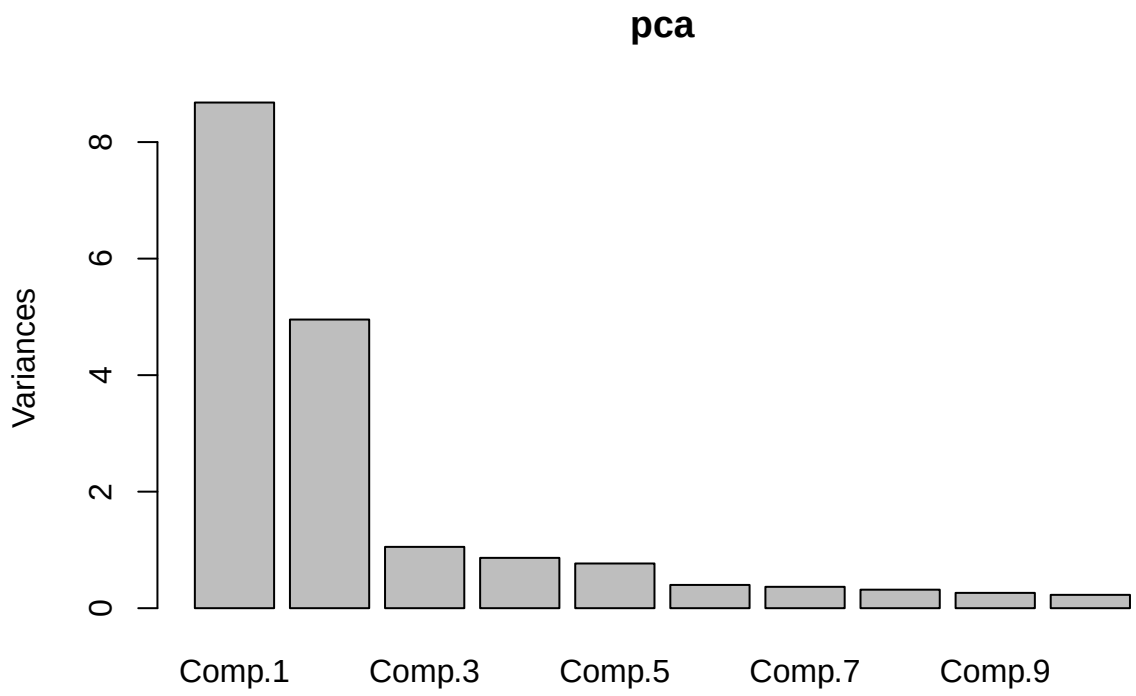
```
## Proportion of Variance 0.0005293647 0.0005209456 0.0005068368 0.0005030104
## Cumulative Proportion  0.9767066935 0.9772276391 0.9777344759 0.9782374863
##                            Comp.75      Comp.76      Comp.77      Comp.78
## Standard deviation     0.1015509833 0.1008202818 0.1003858381 0.0998311357
## Proportion of Variance 0.0004821432 0.0004752297 0.0004711429 0.0004659505
## Cumulative Proportion  0.9787196294 0.9791948591 0.9796660020 0.9801319525
##                            Comp.79     Comp.80      Comp.81      Comp.82
## Standard deviation     0.0970713030 0.096681599 0.0956133567 0.0948554804
## Proportion of Variance 0.0004405442 0.000437014 0.0004274102 0.0004206613
## Cumulative Proportion  0.9805724967 0.981009511 0.9814369209 0.9818575823
##                            Comp.83      Comp.84      Comp.85      Comp.86
## Standard deviation     0.0946714286 0.0939952905 0.0924790696 0.0918238766
## Proportion of Variance 0.0004190305 0.0004130665 0.0003998477 0.0003942022
## Cumulative Proportion  0.9822766127 0.9826896792 0.9830895269 0.9834837291
##                            Comp.87     Comp.88      Comp.89      Comp.90
## Standard deviation     0.0893217774 0.089257379 0.0879328543 0.0878895166
## Proportion of Variance 0.0003730117 0.000372474 0.0003615015 0.0003611452
## Cumulative Proportion  0.9838567408 0.984229215 0.9845907163 0.9849518616
##                            Comp.91      Comp.92      Comp.93      Comp.94
## Standard deviation     0.0867809688 0.0865714520 0.0851138186 0.0848290290
## Proportion of Variance 0.0003520925 0.0003503944 0.0003386943 0.0003364316
## Cumulative Proportion  0.9853039540 0.9856543484 0.9859930427 0.9863294743
##                            Comp.95      Comp.96      Comp.97      Comp.98
## Standard deviation     0.0838186602 0.0826239908 0.0817049255 0.0809671832
## Proportion of Variance 0.0003284651 0.0003191685 0.0003121075 0.0003064967
## Cumulative Proportion  0.9866579393 0.9869771079 0.9872892154 0.9875957121
##                            Comp.99     Comp.100     Comp.101     Comp.102
## Standard deviation     0.0802433963 0.0794694127 0.0786168066 0.0780133904
## Proportion of Variance 0.0003010415 0.0002952621 0.0002889605 0.0002845418
## Cumulative Proportion  0.9878967536 0.9881920157 0.9884809763 0.9887655180
##                           Comp.103     Comp.104     Comp.105     Comp.106
## Standard deviation     0.0779162124 0.0772036247 0.0762127796 0.0760266046
## Proportion of Variance 0.0002838333 0.0002786655 0.0002715585 0.0002702333
## Cumulative Proportion  0.9890493514 0.9893280168 0.9895995753 0.9898698086
##                           Comp.107    Comp.108     Comp.109     Comp.110
## Standard deviation     0.0754600119 0.074074330 0.0737992484 0.0734649174
## Proportion of Variance 0.0002662205 0.000256533 0.0002546312 0.0002523293
## Cumulative Proportion  0.9901360291 0.990392562 0.9906471933 0.9908995226
##                           Comp.111     Comp.112     Comp.113     Comp.114
## Standard deviation     0.0729416810 0.0720596168 0.0717869767 0.0714430478
## Proportion of Variance 0.0002487478 0.0002427681 0.0002409345 0.0002386315
## Cumulative Proportion  0.9911482705 0.9913910386 0.9916319731 0.9918706046
##                           Comp.115     Comp.116     Comp.117     Comp.118
## Standard deviation     0.0703019158 0.0696973470 0.0689548743 0.0684021902
## Proportion of Variance 0.0002310692 0.0002271121 0.0002222991 0.0002187498
## Cumulative Proportion  0.9921016738 0.9923287859 0.9925510850 0.9927698348
##                           Comp.119     Comp.120     Comp.121    Comp.122
## Standard deviation     0.0678123894 0.0675109121 0.0663417012 0.065857371
## Proportion of Variance 0.0002149938 0.0002130864 0.0002057695 0.000202776
## Cumulative Proportion  0.9929848286 0.9931979149 0.9934036844 0.993606460
##                           Comp.123    Comp.124     Comp.125     Comp.126
## Standard deviation     0.0654266783 0.064769636 0.0643689521 0.0634974852
## Proportion of Variance 0.0002001324 0.000196133 0.0001937138 0.0001885041
## Cumulative Proportion  0.9938065928 0.994002726 0.9941964397 0.9943849438
```

```
##                            Comp.127     Comp.128     Comp.129     Comp.130
## Standard deviation     0.0628417159 0.0626617018 0.0618450440 0.0612420106
## Proportion of Variance 0.0001846307 0.0001835744 0.0001788206 0.0001753503
## Cumulative Proportion  0.9945695744 0.9947531488 0.9949319694 0.9951073198
##                            Comp.131     Comp.132     Comp.133     Comp.134
## Standard deviation     0.0610495180 0.0602726726 0.0598603275 0.0590165988
## Proportion of Variance 0.0001742498 0.0001698434 0.0001675274 0.0001628381
## Cumulative Proportion  0.9952815695 0.9954514129 0.9956189403 0.9957817785
##                            Comp.135     Comp.136     Comp.137     Comp.138
## Standard deviation     0.058959282 0.0582037042 0.0575064364 0.0571410692
## Proportion of Variance 0.000162522 0.0001583832 0.0001546111 0.0001526527
## Cumulative Proportion  0.995944300 0.9961026836 0.9962572947 0.9964099474
##                            Comp.139     Comp.140     Comp.141     Comp.142
## Standard deviation     0.0566058710 0.0561097703 0.0559422172 0.0552800844
## Proportion of Variance 0.0001498065 0.0001471922 0.0001463144 0.0001428713
## Cumulative Proportion  0.9965597539 0.9967069461 0.9968532605 0.9969961319
##                            Comp.143    Comp.144     Comp.145     Comp.146
## Standard deviation     0.054814515 0.054325369 0.0536547122 0.0521486754
## Proportion of Variance 0.000140475 0.000137979 0.0001345933 0.0001271435
## Cumulative Proportion  0.997136607 0.997274586 0.9974091792 0.9975363227
##                            Comp.147     Comp.148     Comp.149     Comp.150
## Standard deviation     0.0519291619 0.0512557251 0.0509145857 0.0507118008
## Proportion of Variance 0.0001260754 0.0001228266 0.0001211971 0.0001202336
## Cumulative Proportion  0.9976623981 0.9977852247 0.9979064218 0.9980266554
##                            Comp.151     Comp.152     Comp.153     Comp.154
## Standard deviation     0.0497063618 0.0488269116 0.0478752944 0.0472645515
## Proportion of Variance 0.0001155132 0.0001114619 0.0001071595 0.0001044429
## Cumulative Proportion  0.9981421686 0.9982536305 0.9983607900 0.9984652328
##                            Comp.155     Comp.156     Comp.157     Comp.158
## Standard deviation     0.0464289313 0.0458769984 4.520495e-02 0.0445911931
## Proportion of Variance 0.0001007825 0.0000984006 9.553877e-05 0.0000929621
## Cumulative Proportion  0.9985660153 0.9986644159 9.987600e-01 0.9988529168
##                            Comp.159     Comp.160     Comp.161     Comp.162
## Standard deviation     4.395960e-02 4.316287e-02 4.233672e-02 4.147020e-02
## Proportion of Variance 9.034731e-05 8.710206e-05 8.379962e-05 8.040445e-05
## Cumulative Proportion  9.989433e-01 9.990304e-01 9.991142e-01 9.991946e-01
##                            Comp.163     Comp.164     Comp.165     Comp.166
## Standard deviation     3.994379e-02 3.890809e-02 3.840105e-02 3.740219e-02
## Proportion of Variance 7.459441e-05 7.077624e-05 6.894359e-05 6.540361e-05
## Cumulative Proportion  9.992692e-01 9.993399e-01 9.994089e-01 9.994743e-01
##                            Comp.167     Comp.168     Comp.169     Comp.170
## Standard deviation     0.0365687504 3.568198e-02 3.367672e-02 3.204147e-02
## Proportion of Variance 0.0000625213 5.952586e-05 5.302336e-05 4.799904e-05
## Cumulative Proportion  0.9995368094 9.995963e-01 9.996494e-01 9.996974e-01
##                            Comp.171     Comp.172     Comp.173     Comp.174
## Standard deviation     0.0306268783 2.973992e-02 2.915757e-02 2.768987e-02
## Proportion of Variance 0.0000438544 4.135113e-05 3.974754e-05 3.584674e-05
## Cumulative Proportion  0.9997412121 9.997826e-01 9.998223e-01 9.998582e-01
##                            Comp.175     Comp.176     Comp.177     Comp.178
## Standard deviation     2.443894e-02 0.0227587858 2.214726e-02 2.058651e-02
## Proportion of Variance 2.792368e-05 0.0000242162 2.293232e-05 1.981404e-05
## Cumulative Proportion  9.998861e-01 0.9999102974 9.999332e-01 9.999530e-01
##                            Comp.179     Comp.180     Comp.181
## Standard deviation     1.907157e-02 1.813142e-02 1.766009e-02
```

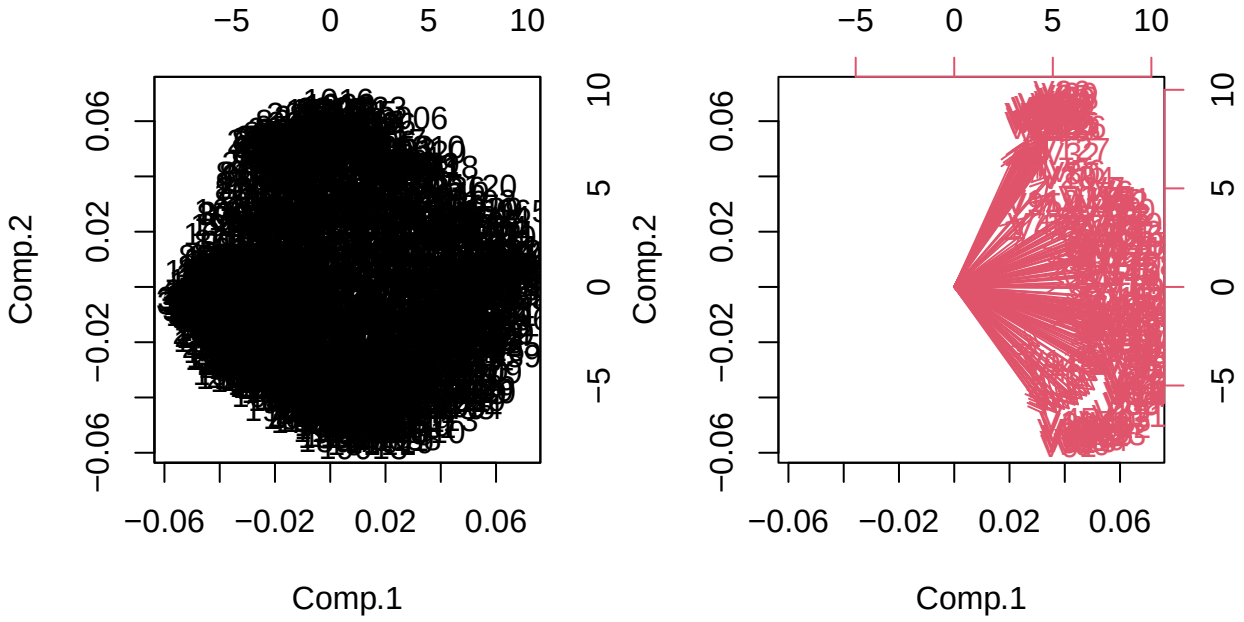
```
## Proportion of Variance 1.700516e-05 1.536991e-05 1.458121e-05
## Cumulative Proportion  9.999700e-01 9.999854e-01 1.000000e+00
```

By observing the screeplot we could select the two first components for representing most of the information.

```
screeplot(pca)
```




```
par(mfrow=c(1,2))
biplot(pca,col=c(1,0))
biplot(pca,col=c(0,2))
```



As we can see with PCA is difficult to clearly visualize the name of the variables. However, we think that the stations following on the right-hand side of the `Comp.1` they are going to be more solicited. As the variables are the hourly values, maybe we should think about plotting them as a TimeSeries once we have done the clustering.

Apply HC

On the hourly loading data we apply HC.

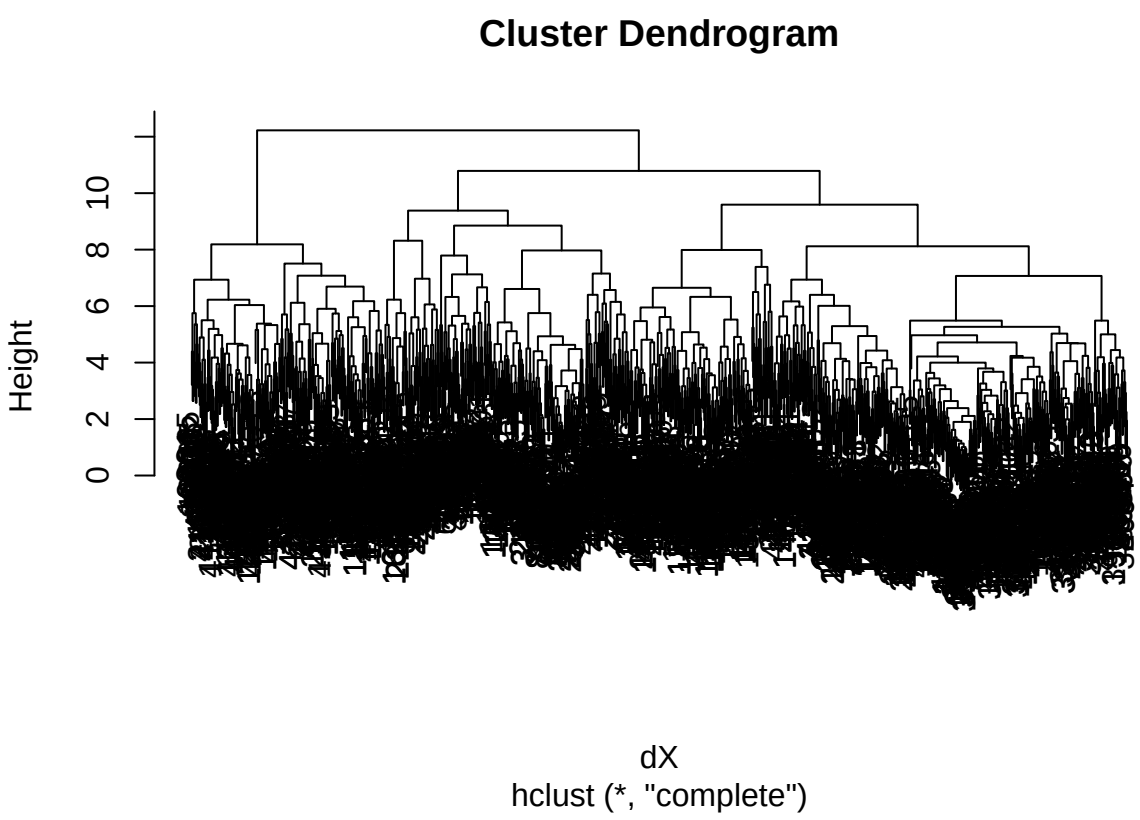
```
gps = as.matrix(data[["position"]])
dX = dist(X)
```

We compare all the distance methods

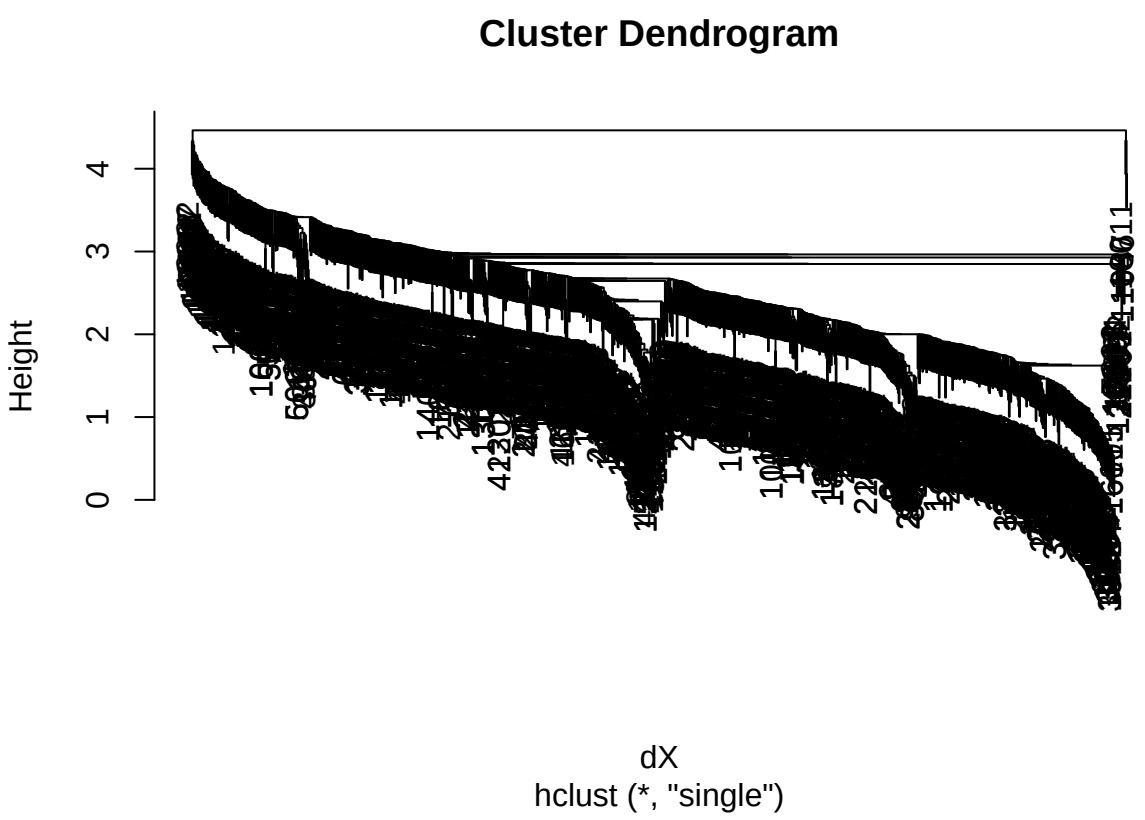
```
hc.compX = hclust(dX,method='complete')
hc.singleX = hclust(dX,method='single')
hc.centroidX = hclust(dX,method='centroid')
hc.wardX = hclust(dX,method='ward.D2')
```

We observe that the complete and the ward distances give balanced hierarchies. Particularly for complete and ward, we select $k = 4$ as it is the largest gap.

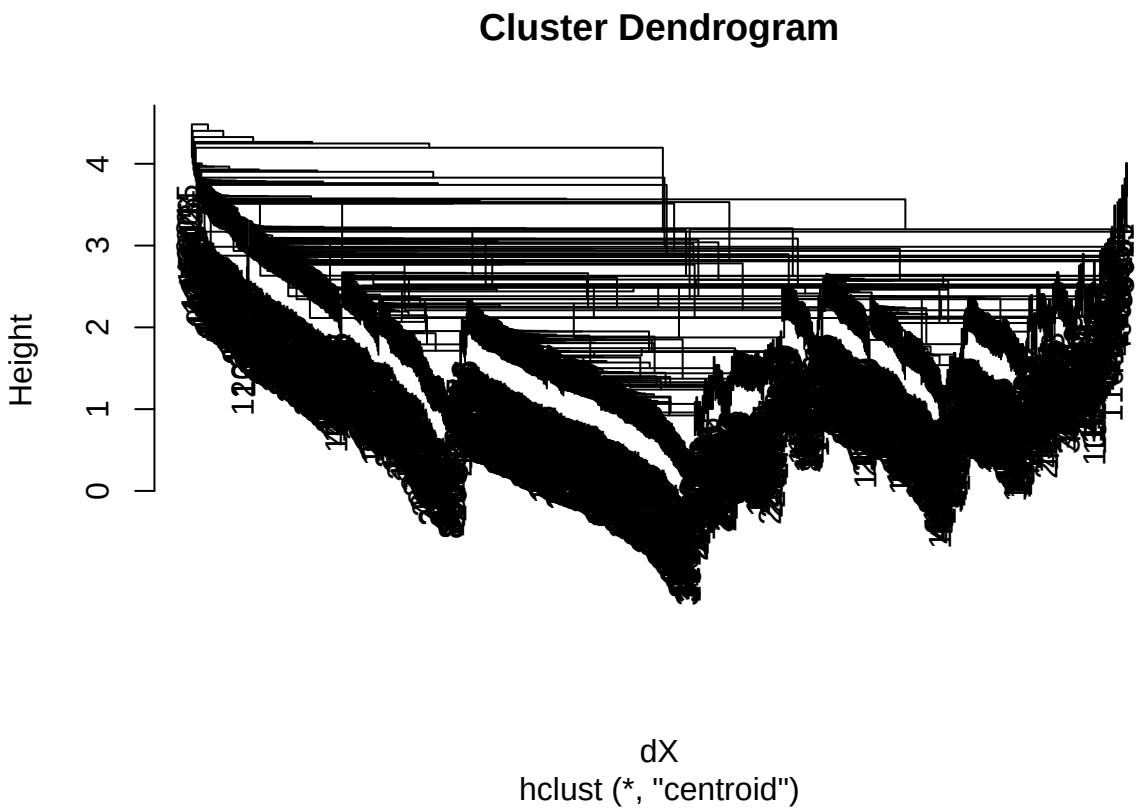
```
plot(hc.compX)
```



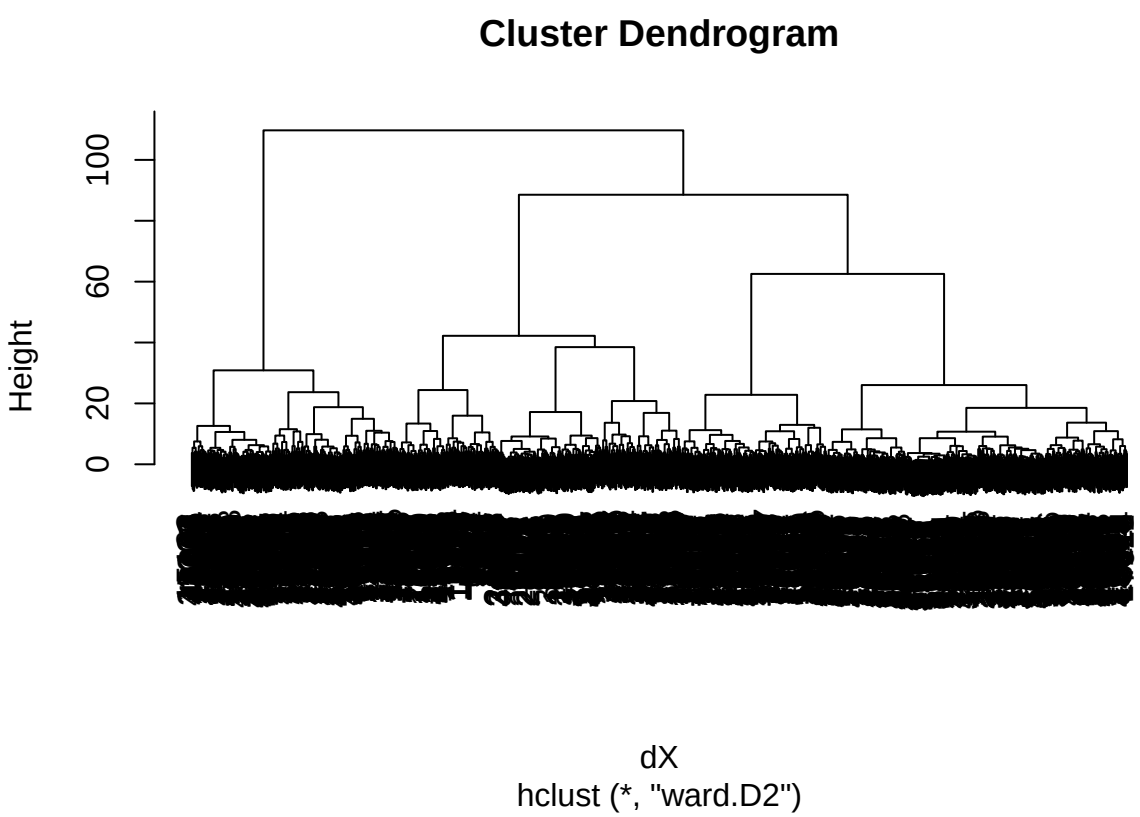

```
plot(hc.singleX)
```




```
plot(hc.centroidX)
```




```
plot(hc.wardX)
```



We can see that the division is made along the river and where the different areas are separated by its proximity.

```
library(leaflet)
plot_map <- function(cluster_obj)
{
  palette = colorFactor("RdYlBu", domain = NULL)
  leaflet(data[["position"]]) %>% addTiles() %>%
  addCircleMarkers(radius = 3,
                   color = palette(cluster_obj),
                   stroke = FALSE, fillOpacity = 0.9) %>%
    addLegend(colors = palette(sort(unique(cluster_obj))), labels = sort(unique(cluster_obj)), position
}
```

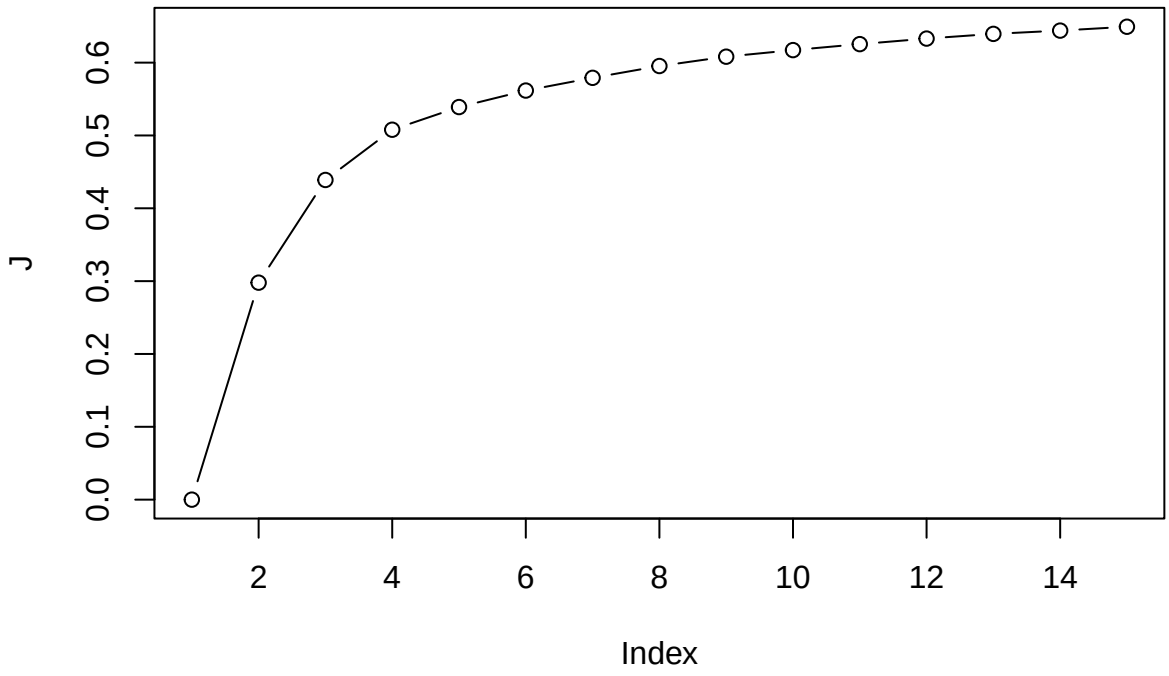
```
clustersX = cutree(hc.wardX, k = 4)
#install.packages("leaflet")
plot_map(clustersX)
```

Apply k-means

By applying k-means to the hourly loads by station, we select $k = 4$ on the screeplot.

```
K.max = 15
J = rep(NA,K.max)
for (k in 1:K.max){
  out = kmeans(X,k,nstart=10) # nstart=10 permits to initialise evaluating between 10 random points.
  J[k] = out$betweenss / out$totss
```

```
}
plot(J,type='b')
```



We can decide to take $k = 4$ as the number of clusters.

We are going to plot the mean of the station's load for each cluster, this could be understood as the average usage for the stations associated with its cluster number.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.4     v stringr 1.5.1
## v forcats 1.0.0     v tibble  3.2.1
## v purrr   1.0.4     v tidyr   1.3.1
## v readr   2.1.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
plot_clusters <- function(km.clus_obj)
{
  cluster_names = sort(unique(km.clus_obj$cluster))
  ccol = palette(cluster_names)

  df = data.frame(t(km.clus_obj$centers))
  colnames(df) <- cluster_names
```

```
  # treat df to ease plotting
dd <- df %>%
  mutate(x_time = h) %>%
  pivot_longer(cols = cluster_names, names_to = "cluster", values_to = "bike station capacity")

ggplot(data = dd, aes(x = x_time, y = `bike station capacity`, colour = cluster)) +
  ylim(c(0,1)) +
  geom_line(lwd = 1) +
  scale_colour_manual(values = ccol) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%a") +
  theme(panel.background = element_rect(fill = "grey"))
}
```

```
png(width = img_width, height = img_height,'4-cluster-means.png')
set.seed(666)
km.clus = kmeans(X,4)
plot_clusters(km.clus)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(cluster_names)
##
##   # Now:
##   data %>% select(all_of(cluster_names))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
dev.off()
```

```
## pdf
##   2
```

When plotting the cluster means we observe that on:

- *cluster 1*: During the **week days** the stations are full at noon during the week days, this means that the stations are close to **business district**. The stations are empty at midnight. During the **weekend** the use is less abrupt. We can identify it in the map as the **city center** in `red`.
- *cluster 2*: During **week days**, it can be seen as the opposite of cluster 1. They are full at midnight and empty at noon, people leave their houses in the morning and start taking bikes. In the evening they come back home and the bike stations start to fill up. We can identify it in the map as the **residential areas** in `orange`. During the **weekend** the peaks are delayed as people come back home late at night.
- *cluster 3*: the stations are most of the time full with peaks at midnight and valleys at noon. They follow the *cluster 2* trend and its trend is steady even during the weekends. This means that this cluster is a **mix of a residential area and a commercial area** in `cyan`.
- *cluster 4*: the stations are most of the time empty, so we can say that these stations are rarely refilled and mostly refilled in the morning very soon, from midnight to 6 am. During the weekends they are mostly empty. This behaviour invites to think that these stations are mostly for consuming bikes and rarely as destination and they are refilled very soon in the morning, probably by Velib workers. You can find them in `dark blue` on the map.

Then, we plot the stations classified by cluster on the map, we see that the stations are mixed all-around the

city.

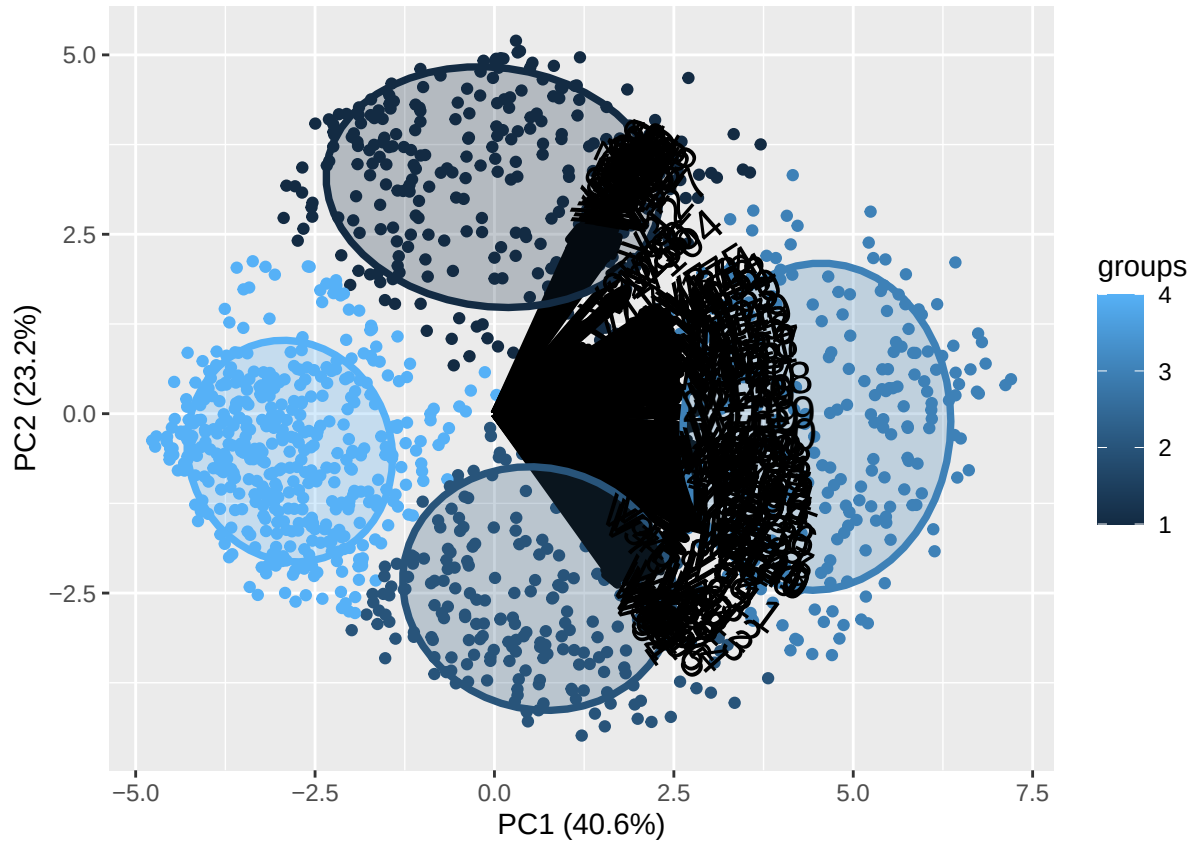
```
plot_map(km.clus$cluster)
```

We obtain very similar results to the HC with ward distance and $k = 4$.

PCA + k-means

If we plot the 4 clusters projected on the two main component data we obtain the following plot.

```
#install.packages("ggbiplot")
library(ggbiplot)
arrow_col = "black"
pca_plot <- ggbiplot(pca,
         obs.scale = 1, var.scale = 1,
         varname.size = 6,
         groups = km.clus$cluster,
         varname.adjust = 1.25,
         varname.color = arrow_col,
         choices = 1:2, # components 1 and 2
         ellipse = T, circle = F)
pca_plot
```



We observe that the 2 main components project in a clear way the 4 clusters, creating a squared shape divided in 4 parts of almost the same size. Remark that that cluster 1 (top) and 2 (bottom), corresponding to city center and residential areas respectively, are opposite in the graph. The variables vectors are all of the same length and they have an spectrum that sweeps the 4th and 1st quadrants of the graph. Each variable vector correspond to an hour of the day, we assume that weekdays and weekend hours are different as they shown

different behaviours. We know that the clusters are going to be assinged propotionally to the direction of the arrows, i.e. if an arrow is pointing to a cluster (positively related), during this hours the stations are going to be filled (bikes arriving to these stations). Inversely, if an arrow is opposite to the hour, the bike stations are going to be emptied (bikes departing from these stations).

However, this plot is not yet clear enough. Let's plot for a given day, monday (day = 1), the hours associated with noon and midnight. Also, let's change the color of the hourly variables (arrows) to better show what happens at **noon** and at **midnight** for each cluster.

We have created a function to substitute the colors of the arrows and labels on the `ggbiplot` object.

```
biplot_arrows <- function(ggbiplot_obj, day_to_plot, from_hours, to_hours, col_vec)
{
  g <- ggplot_build(ggbiplot_obj)
  # change the colour of the arrows according to its hour
  if ( length(from_hours) == length(to_hours)
       & length(from_hours) == length(col_vec) ) {

    for (i in 1:length(from_hours)) {
      # change to new colour
    cond = (hour(h) >= from_hours[i] & hour(h) <= to_hours[i])
    morning = g[["data"]][[2]][["colour"]][cond]
    mor_col = col_vec[i]
    # replace arrows
    g[["data"]][[2]][["colour"]][cond] <- replace(morning, morning==arrow_col,mor_col)
    # replace arrow labels
    morning = g[["data"]][[4]][["colour"]][cond]
    g[["data"]][[4]][["colour"]][cond] <- replace(morning, morning==arrow_col,mor_col)
    }
  # plot only variable arrows from day = 1
  cond = day(h) == day_to_plot # condition vector on time
  g$data[[2]] <- g$data[[2]][cond,]
  g$data[[4]] <- g$data[[4]][cond,]

  # Repackage and plot
  plot(ggplot_gtable(g))
  }
  else {
    stop("from_hours and to_hours and col_vec vectors must have the same length")
  }
}
```

For a weekday, the biplot looks like this.

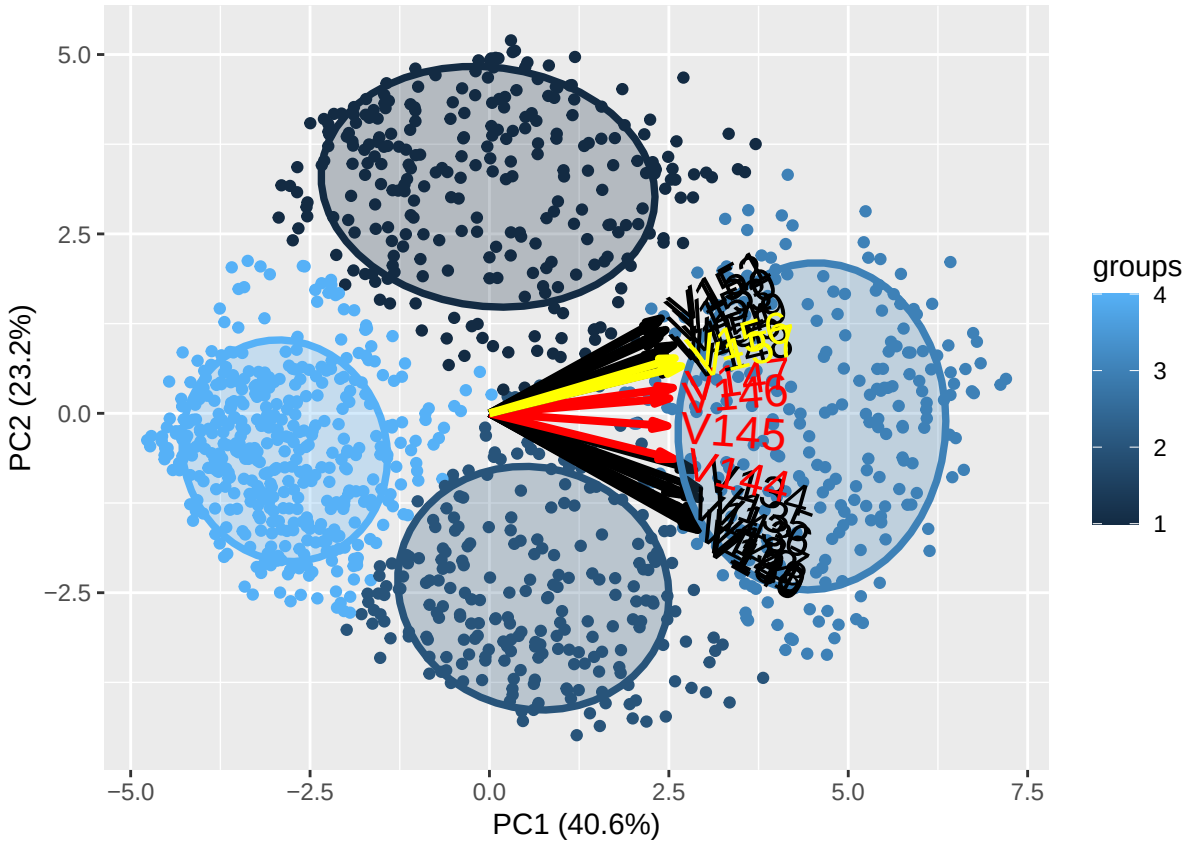
```
png(width = img_width, height = img_height, "pca-circle.png")
VAR_DAY = 1 # monday = 1, sunday = 7
biplot_arrows(ggbiplot_obj = pca_plot,
              day_to_plot = VAR_DAY,
              from_hours = c(10, 22),
              to_hours = c(13, 24),
              col_vec = c("red","yellow")
              )
dev.off()
```

```
## pdf
##   2
```

We observe that at **noon** (10h-13h) (red colour), the stations in cluster 1 (city center) are a going to be a destination (full with bikes). We observe that at **midnight** (22h-24h) (yellow colour), the stations in cluster 2 and 3 are going to be a destination.

For a weekend, we plot aswell the same hours

```
VAR_DAY = 6 # monday = 1, sunday = 7
biplot_arrows(ggbiplot_obj = pca_plot,
              day_to_plot = VAR_DAY,
              from_hours = c(10, 22),
              to_hours = c(13, 24),
              col_vec = c("red","yellow")
              )
```



During saturday (weekend) the noon and midnight hour vector change and are assigned to cluster 3, which is the mixed one and also the one that a higher use during the weekend (see TimeSeries plot on Apply k-means section).

Summary

We have used PCA to reduce the dimension to two main components that explain almost half of the information and to obtain a visualization. We observed that with 19 components we could explain 90% of the information contained in the hourly bike use.

We have used HC and k-means to cluster 4 groups depending on the loads of the bike stations.

We have combined the clustering with PCA to show the effect of each variable on the clustering. We have modified the biplot graph to better show the relationship between variables and clustering attribution.

We have plotted the cluster means (mean average capacity) of each bike station to better understand its

usage during a week.